

<<Go语言·云动力>>

图书基本信息

书名：<<Go语言·云动力>>

13位ISBN编号：9787115283078

10位ISBN编号：7115283079

出版时间：2012-6

出版单位：人民邮电出版社

作者：[新加坡] 樊虹剑

页数：294

字数：250000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

Less is more. (舍既是得。

) --Robert Browning 从1978年《C程序设计语言》问世到现在，计算机世界已经发生了翻天覆地的变化。

计算机本身的性能已经得到几千倍的提高，计算机的互联互通更使很多不可思议的事情成为常态。身为计算机灵魂的工程师，程序员一直处在这场变革的风口浪尖。

他们既有弄潮儿的自信与洒脱，又时刻面临被后浪吞噬的危险。

更微妙的是，尽管计算机科学的先驱们很早就发现，编程是门文字和思维的艺术，而且顶尖程序员的生产力要远大于集体化大规模生产作业中的员工。

但是，由于整个计算机产业的迅速发展，以及具备艺术细胞的程序员的稀缺，从业者只能重点投资工具和流程。

自从20世纪60年代提出“软件危机”这一问题以来，各种理论和实践不断涌现，而通过变革编程语言来解决问题的努力，也一直在继续。

但到目前为止，软件还在危机，同志仍需努力。

C.A.R Hoare 在1980年图灵奖获奖致辞中阐述得很精辟：“设计软件有两种方式：第一种是尽量简化，使之明显地看不到弱点；另一种是尽量复杂，从而看不到明显的弱点。

第一种方式非常困难。

就像发现复杂的自然现象之中蕴含着简单的物理定律一样，它需要投入、明察、甚至灵感。

“今天，我非常高兴地向大家介绍Go语言。

40年前，C编程语言极大地解放了程序员的生产力。

今天，新兴的Go语言也一定能够再次提升程序员的创造力和协作能力，从而让有才能的程序员得以在创作中充分发挥，让程序员重新成为令人尊敬的计算机灵魂的工程师。

目前，Go语言才发布了短短几年时间，就已经赢得了广泛赞誉，并得到了广泛应用。

它不但用于支持谷歌和Heroku的云计算平台，还在一些大公司的内部系统中担当关键角色，甚至普通程序员也用Go代替其他常见语言来处理手边的问题。

这充分说明Go是一种成熟、实用、好用的语言。

它的未来非常光明。

本书所介绍的内容正是使用Go语言时必不可少的基础知识。

作者不满足于平铺直叙的方式，而希望能通过轻松的表达，让Go自然而然地接近读者，并以作者自身的经验和观点，启发读者独立地分析和试验，从而让读者能顺利地入门到精通，让Go成为最好的编程语言。

<<Go语言·云动力>>

内容概要

Go语言是由谷歌的Rob Pike、Ken Thompson和Robert Griesemer共同设计开发的一种新型程序设计语言。

2012年Go 1的推出，代表着Go语言的稳定成熟，也正式宣告Go走入了主流语言的行列。

《Go语言·云动力》是Go语言程序设计入门书，介绍了Go语言的基础知识，包括静态类型、流程控制、函数、动态类型、面向对象、并发编程等内容，以及同其他C类语言相比，Go所具备的全新特性。

同时，本书还介绍了Go语言在云计算中的应用。

《Go语言·云动力》适合Go语言初学者学习。

作者简介

樊虹剑

15岁迷上Apple II，立志成为程序员。

从Applesoft BASIC，到Microsoft C、C++、C#，再到Apple Objective-C，其间使用过大小十几种编程语言，始终不得要领。

不明白起源于数学和电子学的计算机科学，为何不去追求诗歌般至简至纯的美学，而是纠结于繁复的形式和空洞的哲学。

几欲放弃。

直到偶然发现Plan 9，才获得返璞归真的乐趣，并一路欢喜走过Inferno，快乐走入Go语言的世界。

<<Go语言·云动力>>

书籍目录

目 录

- 第1章 快速入门 1
 - 1.1 编辑和编译 2
 - 1.2 世界，你好！
9
 - 1.3 自我复制 11
 - 1.4 猜数游戏 14
 - 1.5 图灵机 19
 - 1.6 排版工具 23
 - 1.7 游乐场 30
 - 1.8 位钱 36
 - 1.9 小结 43
- 第2章 静态类型 44
 - 2.1 数制 45
 - 2.1.1 整数 45
 - 2.1.2 二进制补码 46
 - 2.1.3 整数比较 47
 - 2.1.4 按位运算 48
 - 2.1.5 浮点数 50
 - 2.1.6 复数 51
 - 2.1.7 优先级 52
 - 2.1.8 表达式 53
 - 2.2 字符和字符串 54
 - 2.2.1 UTF-8编码 54
 - 2.2.2 Unicode字符 55
 - 2.2.3 转义字符 56
 - 2.2.4 字符串 56
 - 2.2.5 字符串转换 57
 - 2.3 数组 60
 - 2.3.1 声明 60
 - 2.3.2 下标 61
 - 2.3.3 赋值 61
 - 2.4 切片 63
 - 2.5 结构体 67
 - 2.5.1 项 68
 - 2.5.2 内置 69
 - 2.6 指针 71
 - 2.7 小结 73
- 第3章 流程控制 74
 - 3.1 简单语句 75
 - 3.2 判断语句if 77
 - 3.3 多分支语句switch 79
 - 3.4 循环语句for 82
 - 3.5 遍历 84
 - 3.6 标号和跳转 85

<<Go语言·云动力>>

- 3.7 作用域 86
- 3.8 小结 88
- 第4章 函数 89
 - 4.1 签名 90
 - 4.2 参数 91
 - 4.3 返回语句 93
 - 4.4 函数调用 95
 - 4.5 闭包 97
 - 4.6 压后 99
 - 4.7 派错和恢复 100
 - 4.8 方法 102
 - 4.9 包 105
 - 4.10 导入 107
 - 4.11 程序初始化 108
 - 4.12 小结 110
- 第5章 动态类型 111
 - 5.1 映射 112
 - 5.2 界面类型 116
 - 5.3 界面值 119
 - 5.4 error界面 122
 - 5.5 有界无类 124
 - 5.6 排序 126
 - 5.7 类型断言 130
 - 5.8 类型分支 133
 - 5.9 反射 135
 - 5.10 小结 141
- 第6章 面向对象 142
 - 6.1 背景 143
 - 6.2 术语 146
 - 6.3 与C++对比 149
 - 6.3.1 继承 149
 - 6.3.2 抽象类 151
 - 6.3.3 泛型 153
 - 6.4 小结 154
- 第7章 并发编程 155
 - 7.1 背景 156
 - 7.2 同步通信 158
 - 7.3 去程 162
 - 7.4 程道 164
 - 7.5 遍历与关闭 166
 - 7.6 MapReduce 168
 - 7.7 select语句 170
 - 7.8 程道值 173
 - 7.9 互斥 175
 - 7.10 小结 177
- 第8章 云计算 178
 - 8.1 背景 179

<<Go语言·云动力>>

- 8.2 GAE 181
- 8.3 Hello 世界！
183
- 8.4 画胡子 185
- 8.5 留言录 191
- 8.6 用户API 195
- 8.7 数据库API 196
 - 8.7.1 术语 197
 - 8.7.2 Go数据库API 197
 - 8.7.3 实体键 199
 - 8.7.4 查询和索引 200
 - 8.7.5 实体组 201
 - 8.7.6 限制 201
- 8.8 交易 203
- 8.9 散段 205
- 8.10 内存缓冲 208
- 8.11 大件库 210
- 8.12 URL抓取 213
- 8.13 任务队列 214
 - 8.13.1 任务 216
 - 8.13.2 任务执行 217
 - 8.13.3 队列 218
- 8.14 后端 220
- 8.15 能力API 221
- 8.16 电子邮件API 222
 - 8.16.1 发送 222
 - 8.16.2 接收 223
- 8.17 信道API 224
- 8.18 小结 226
- 第9章 标准包 227
 - 9.1 格式包 228
 - 9.1.1 格式输出Printf 229
 - 9.1.2 动词表 230
 - 9.1.3 宽度和精度 231
 - 9.1.4 报错 232
 - 9.1.5 额外标记 232
 - 9.1.6 格式输入 233
 - 9.1.7 字符串格式 234
 - 9.2 字节包 236
 - 9.3 模板包 239
 - 9.4 正则表达式包 245
 - 9.5 时间包 253
 - 9.6 超链接包 258
 - 9.6.1 http服务器和客户机 258
 - 9.6.2 https加密通信 260
 - 9.6.3 Get 263
 - 9.6.4 Post 265

9.6.5	Cookie	268
9.7	编码包	271
9.7.1	gob	271
9.7.2	json	275
附录A	Go的安装和使用	277
附录B	EBNF	284
附录C	中英术语对照表	291

章节摘录

第1章 快速入门 Well begun is half done. (万事开头难。

) -- 谚语 本章通过几个具体的案例, 解释Go语言程序的基本结构, 让有一定功力的读者迅速上手, 也让初学者对Go有个初步印象, 便于后续章节的展开。

1.1 编辑和编译 大家都知道, 学习编程的最佳方式就是动手编程。

但这里有一个巨大的障碍。

动手之前你要熟悉相关的工具, 知道怎样使用编辑器, 写出源代码。

怎样操作编译器、链接器, 得到可以执行的程序。

另外还要考虑怎样运行这个程序, 去哪里查看运行输出。

并且还要知道, 每一个步骤如果出了错, 该怎样应对。

本书不打算手把手地教你使用这些必要工具。

最好的办法是请教身边的专家。

也可以参考其他资料, 尤其是Go的正式发布网站www.golang.org, 自己慢慢摸索, 并在使用中不断尝试。

为了帮助初学者克服这些与语言无关的技术障碍, 我们特意整合了Windows上的Acme程序编辑器和Go编译器, 并针对Go的编写、编译、执行和除错做了一个简单演示。

Go也可以运行在Linux、FreeBSD和Mac OS X上。

读者可以根据自己的使用环境和习惯相应地调整。

这些平台上Go语言环境的安装和使用, 请参考附录。

读者可以从本书的支持网站www.goplace.org下载acme.zip。

它完全不需安装, 直接解压到Windows的c:\即可。

Go工具和运行环境要求安装在c:\Go下面。

而Acme是在c:\acme.app下面, 执行它里面的acme.bat, 就可以启动编辑器了。

我们看一下acme.bat: set GOROOT=c:\Go set GOPATH=c:\acme-home\PATH=%PATH%;%GOROOT%\bin c: cd \acme.app\Contents\Resources acme.exe 这里有几个重要的路径: “GOROOT是Go的正式软件所在路径; “GOPATH是我们自己的软件和第三方的Go软件所在路径; “PATH中必须包括GOROOT的bin才可以使用Go工具; “acme-home是Acme存放文件的根目录。

如果读者希望安装到其他路径下, 则需要对acme.bat做相应的修改。

Acme很神秘?

其实编辑器就像钢笔铅笔, 纯属个人喜好。

一开始用哪个都写不出好字。

慢慢地习惯了一种, 就觉得它用着顺手了。

所以, 还没有习惯使用具体某一款编辑器的读者, 不妨多试试, 或许你会发现Acme是最出色的程序编辑器。

顺便一提, Acme和Go一样, 是Rob Pike的作品。

C语言和Unix之父Dennis Ritchie也使用Acme。

启动Acme后的第一感觉就是”它很不一般“, 而且几乎也可以说是”不知所措“。

菜单在哪里?

帮助在哪里?

怎么打开文件?

在直接给出这些答案之前, 我们有必要了解它的设计哲学。

只有明白了我们真正需要的是什么, 才晓得为何这样以及如何才能这样。

Acme和Unix的设计哲学是一脉相承的, 都是只提供少量基本工具以及组合它们的方法, 而不是针对每一项需求都准备互不相干的几十个上百个选项。

<<Go语言·云动力>>

其实，Acme就是一个运行在类似Unix的虚拟机上的编辑器。

这个虚拟机使用Inferno操作系统。

这和Emacs编辑器运行在Lisp虚拟机上类似。

只不过，Emacs只能使用Lisp编程来配置编辑器的各项功能，而它的Lisp程序只为Emacs服务。

Acme则只是Inferno的一个程序。

我们可以使用类似Unix的sh脚本，以及使用类似Go的Limbo编程语言，来编写Inferno的其他命令和程序，与Acme一起配合，完成所需的编辑工作。

这样的Acme，也可以认为是Inferno操作系统的图形用户界面（GUI）。

只不过，相对于传统WIMP（窗口、图标、菜单、指针）的GUI风格，Acme简化为WP（窗口和指针）--刚好也可以代表Word Processing（文字处理）。

这是因为，与其在几千年后再来发明一套类似象形文字的图标，为何不直接使用人们已经使用了很久的字词呢？

与其干等着别人为你做好菜，让你伸手去点单，为何不去自己享受做菜的乐趣呢？

难道程序员的本行不是编程吗？

为何程序员所用的编辑器就不能自己编程定制功能呢？

所以，学会了使用Acme的程序员，得到了这些问题的答案以后，就再也不能容忍传统的编辑器和IDE了。

到底如何才能打开文件？

从哪里能得到帮助？

欢迎Windows世界的同学打开窗口，感受Unix世界的清新和煦。

Unix的世界除了文件，就是文字。

例如，在Acme的随便什么地方打入个斜杠字符“/”--这是Unix的根目录和目录分隔符，再用鼠标右键单击这个字符，就可以看到根目录下的文件和子目录。

Acme的根目录在Windows中是c:\acme.app；而在Acme窗口的最上面一行输入一个点“.”，并同样使用鼠标右键单击这个点，就可以看到用户目录，在Windows中该目录为c:\acme-home。

而Windows的根目录是c:\，这里把它定义为/me。

这个定义可以在根配置文件/lib/sh/profile中看到。

要打开这个文件，当然可以在Acme的任意位置输入文件名再单击右键，也可以从/开始一直右键选择每个目录名，直到所需文件出现在一个窗口中。

文件名所在的浅蓝行是窗口栏，在此处输入me再单击右键，就在profile文件里面找到第一次出现的me。

同样连续单击右键，就一直查找下去。

由此可见，在Acme里查找文件和查找文字都只需用无名指单击即可。

说到窗口，Acme的多窗口是并行排列的。

默认是两列，每列内可以打开多个窗口，每个窗口都有一行天蓝色的标题栏。

滚动条在左侧。

滚动条和标题栏交汇的小方格，在窗口内容有变化时变成深蓝。

按住鼠标左键移动此方格，可以重新排列窗口的位置。

单击可以扩大窗口。

而用中键或右键单击，则可以整列显示此窗口，或再次显示其他窗口的标题栏。

在滚动条内单击鼠标左键和右键可以上下翻页，或用中键单击和移动光标到所需位置。

Acme最独特的是鼠标语言。

它的三个鼠标键经单独或组合使用可以完成大量的操作。

具体操作如表1-1所示。

表1-1 Acme鼠标键组合表	左	中	右	代	表
1+00	移动光标	1++00	选词（整行或括号、引号括起来的段落）		
1--00	选择范围（选中部分高亮显示）	01+0	执行命令	01--0	红色高亮显示选中部分，执行命令
001+	打开文件或查找	001--	绿色高亮显示选中部分，打开或查找		

<<Go语言·云动力>>

1-- 2+ 0 删除选择内容 1-- 2+ 3+ 粘贴回删除的内容, 即复制 1+ 0 2+ 粘贴回删除的内容 2+ 1+ 0 将之前选中的内容传给命令执行 这里, 1、2、3是按键顺序, 0代表不按键。

+是单击, ++是双击, --是按键拖动鼠标。

可以简单地总结为: 左键选取文字, 中键执行, 右键查看选中内容, 左中删, 左右贴。

试着练习几分钟之后, 这些也就成了小脑指令了--无需大脑去想了。

如果没有三键鼠标, 建议你去买一个, 因为你可能以后会经常用到Acme。

当然, 偶尔也可以用普通鼠标的滚轮代替中键, 甚至在笔记本电脑上用Ctrl配合右键代替中键。

或者使用表1-2所示的Ctrl键组合执行常用命令。

表1-2 Acme键盘命令表 Ctrl键组合 执行操作 a 转至行首 b 返回上页 c 复制 d 补全 e 转至行尾 f 转至下页 h 退格 i Tab k 左移 l 右移 m 换行 n 下移 o 上移 p 转至文件尾 q 转至文件头 s 保存 u 删整行 v 粘贴 w 删除 x 剪切 y 重复 z 撤销 习惯Acme的关键就是鼠标命令, 而使用鼠标命令的方便之处在于, Acme的所有文字, 都是可以用鼠标命令来操作的。

也就是说, 不管命令位于何处, 是在标题行或者一个窗口里, 只要在其上按中键或按住中键扫过, 都是执行保存文件的操作。

而在其上按右键, 都是执行查找的操作。

这样, 我们把一些最常用的命令放在每个窗口的标题行, 只要中指轻轻一点就可以了。

这些命令和一些不太常用的命令如表1-3所示。

表1-3 Acme鼠标命令表 中 文 英 文 代 表 剪 Cut 剪切选中的内容 拷 Copy 复制选中的内容 贴 Paste 粘贴上次剪切的内容 新 New 在新窗口中打开选中的文件 关 Del 关闭此窗口 悔 Undo 撤销上次修改 不悔 Redo 重复上次撤销 存 Put 保存文件 读 Get 重新读取文件 查 Look 查找选中的词语或文件 令 Edit 执行选中的编辑命令 编 Compile 编译Go文件 Zerox 在新窗口打开同一文件 Putall 保存所有窗口内容 Dump 保存窗口状态, 下次自动打开 Newcol 增添一列 Delcol 删除一列 Exit 退出Acme (不保存) win 命令控制窗口 man 帮助手册 g 查找C和Go文件的内容 mkdir、touch、rm 新建目录, 新建文件, 删除目录或文件 一些常用的编辑命令可以保存在/guide文件里。

每次选中一条命令, 用鼠标中键加左键单击在某个窗口的““令“”字符上, 就可以对此窗口文件的内容执行编辑操作了。

例如, 选择几行程序, 执行, s/^/g命令就可以在每行首加一个tab字符, 也就是整体右移。

而执行, s/^/g就可以去掉行首的tab, 整体左移。

这些编辑命令, 可能是初学者的最大障碍, 却也是Acme最强大、最灵活的地方。

所有的帮助文件, 都在/man目录下面。

如果要学习Acme的编辑命令, 读者可以在Acme里执行man acme; 如果要看shell的编程, 执行man sh即可; 而右键单击类似sh-std(1)这样的格式, 也可以帮助我们看到帮助文件。

这是因为, 我们在lib/plumbing里, 定义了右键的规则, 让这一点也能具备可以编程的一点智能。

不多讲了。

再次重申, 编辑器的选择是个人喜好, 选择什么都和Go语言无关。

……

<<Go语言·云动力>>

编辑推荐

云计算时代的新型编程语言 编程极客不可不学 示例丰富，引人入胜，一本就Go！

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>